Ingénierie et Conduite de projets numériques

4. Production & Déploiement

Produire, livrer, pérenniser :
passer du “design validé” à la réalité de production, diffusion, conformité, suivi

Production et pilotage

Passage au développement

- **But** : Les maquettes finales et les spécifications validées sont transmises aux développeurs pour commencer la phase de **développement**.
- **Livrables** : Prototypes haute-fidélité, spécifications fonctionnelles, guides de style, composants UI.
- **Collaboration** : Les designers UX travaillent souvent en collaboration avec les développeurs pour s'assurer que la conception est bien implémentée, respectant l'expérience utilisateur et les détails visuels.



1. Phase Nom de domaine

Choix du Nom de Domaine

- **But** : Le nom de domaine est l'adresse de votre site sur Internet (ex. : www.monsite.com). Il doit être unique et facile à retenir.

- **Comment faire** :
  - Allez sur un **registrar de domaines** comme **Gandi**, ou **créer votre nom de domaine chez votre hébergeur**

  - Vérifiez si le nom souhaité est disponible et achetez-le en même temps vous achetez l'hébergement.

- Nom de domaine > identité du projet



2. Phase d'Hébergement

Choix de l'Hébergeur

- **But** : L'hébergeur fournit un espace de stockage sur un serveur pour vos fichiers, afin que votre site soit accessible en ligne.

- **Exemple d'hébergeurs populaires** :
  - **O2switch** (hébergeur français avec une offre tout-en-un),
  - **OVH** (également français, avec plusieurs options d'hébergement),

- **Comment faire** : Choisissez un plan d'hébergement en fonction de vos besoins (nombre de visiteurs prévus, espace de stockage, etc.). Hébergement mutualisé / dédié / cloud

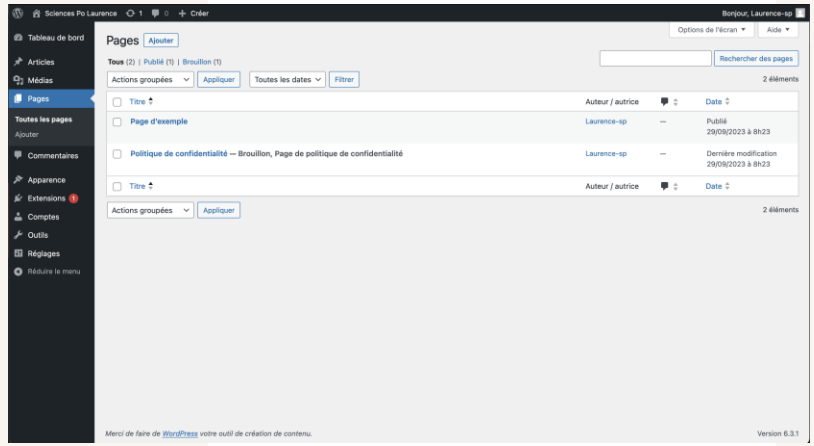




3. Phase de Développement

CMS et Frameworks

- Le **CMS** est souvent choisi pour faciliter la mise en ligne et la mise à jour des contenus : WordPress, Drupal, Joomla, Shopify, Magento
- Mais il faut comparer avec des **Frameworks** plus personnalisés si on a besoin de fonctionnalités uniques : React, Vue, Angular
- Ce choix doit se faire **dès la phase de cadrage**, car il conditionne tout le projet.
- **Critères** : budget, autonomie de l'équipe, évolutivité



3. Phase de Développement

Impact du choix technologique

Chaque choix technique a des conséquences :

- **Budget** (licences, hébergement, compétences nécessaires),
- **Maintenance** (facilité à corriger ou mettre à jour),
- **Sécurité** (plugins ≠ vulnérabilités),
- **Liberté créative** (limites d'un CMS).

C'est pour ça qu'un chef de projet doit comprendre ces enjeux, même sans coder.

3. Phase de Développement

Développement & intégration

- La production, c'est la **traduction des maquettes UX/UI en code et fonctionnalités réelles**.
- Deux logiques :
  - intégration graphique (front-end),
  - développement fonctionnel (back-end).
- Un projet échoue souvent par manque de communication entre design et dev → d'où l'importance de la coordination.
- La réussite dépend de la qualité de l'échange entre les métiers.
- Les designers doivent fournir des maquettes claires, documentées.
- Les développeurs doivent anticiper les contraintes techniques.
- Et tout le monde doit partager une **même vision du livrable attendu**.

3. Phase de Développement

Développement des fonctionnalités dynamiques

- **But** : Développer ou configurer des fonctionnalités dynamiques comme des systèmes de gestion utilisateurs ou des passerelles de paiement.

- **Outils & Plugins :**
  - **WordPress** : WooCommerce (pour l'e-commerce), ACF (Advanced Custom Fields pour la gestion des contenus personnalisés).
  - **Drupal** : Webform (pour les formulaires), Commerce (pour les boutiques en ligne).
  - **Shopify** : Applications et plugins via le Shopify App Store (ex. Oberlo, ReCharge).
  - **Plugins CMS** : SEO plugins (RankMath pour WordPress), plugins de sécurité (WordFence pour WordPress, Sucuri).
  - **Symphony** : Frameworks

4. Phase de Tests

Tests techniques

- Avant de mettre en ligne, il faut tester :
  - Compatibilité multi-supports (mobile, tablette, desktop),
  - Performances (temps de chargement),
  - Bugs.

- C’est ce qu’on appelle la **recette technique**.

4. Phase de Validation & Recette

Tests / Recette

- La recette, c’est la validation finale avant diffusion.
- On passe par une checklist des fonctionnalités, des tests utilisateurs, et une validation du client.
- Le but : éviter les mauvaises surprises en ligne.

5. Suivi et tests post-développement

Tests fonctionnels sur le CMS

- **But** : Tester les fonctionnalités pour vérifier que l'expérience utilisateur fonctionne comme prévu.

- **Outils** :
  - **BrowserStack**, **LambdaTest** : Pour tester la compatibilité sur différents navigateurs et appareils.
  - **Pingdom**, **GTmetrix** : Pour tester la performance du site et les temps de chargement.
  - **SSL Labs** : Pour vérifier la sécurité et les certificats SSL.
  - **WAVE**, **axe** : Pour tester l'accessibilité du site.

WAVE
web accessibility evaluation tool

5. Suivi et tests post-développement

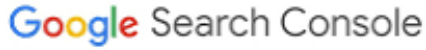
Optimisation pour le SEO et accessibilité

- **But** : Optimiser le site pour les moteurs de recherche (SEO) et assurer sa conformité avec les normes d'accessibilité.

- **Outils** :
  - **RankMath** (pour WordPress) : Optimisation des métadonnées, gestion des balises, analyse SEO.
  - **Yoast SEO** (pour WordPress) : Optimisation des métadonnées, gestion des balises, analyse SEO.
  - **Google Search Console**, **Ahrefs** : Pour surveiller et optimiser le SEO.
  - **WAVE**, **Lighthouse** (outil Google Chrome) : Pour analyser et améliorer l'accessibilité.
  - **Smush**, **ShortPixel** : Pour l'optimisation des images et la performance du site.









6. Lancement et suivi post-lancement

Lancement du site/app via le CMS

- **But** : Mettre en ligne le site web pour qu'il soit accessible aux utilisateurs finaux, et assurer un suivi après le lancement pour garantir une bonne performance et disponibilité.

1. Mise en production du site :

- **Outils** :
  - **CMS (WordPress, Drupal, etc.)** : Dans le CMS, utilisez les options de publication pour mettre le site en ligne.
  - **Action** : Assurez-vous que tous les fichiers, les configurations de base de données, et les paramètres de votre CMS sont bien en place pour que le site soit fonctionnel.

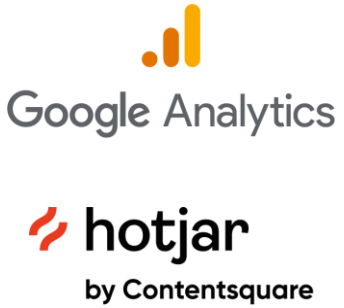
6. Lancement et suivi post-lancement

Lancement du site/app via le CMS

2. Suivi de l'activité du site après le lancement :

- **Outils** :
  - **Google Analytics** : Pour suivre le trafic, le comportement des utilisateurs, et les performances du site.
  - **Hotjar (optionnel)** : Pour obtenir des informations plus qualitatives, comme les cartes de chaleur, afin de voir comment les utilisateurs interagissent avec le site.

- **Action** : Installez les codes de suivi Google Analytics et, si besoin, Hotjar, via votre CMS ou en ajoutant des scripts directement dans le code source.



6. Lancement et suivi post-lancement

Lancement du site/app via le CMS

3. Sauvegardes et maintenance

• **Outils** :
  - **UpdraftPlus (pour WordPress)** ou **Backup & Migrate (pour Drupal)** : Configurez des sauvegardes régulières pour éviter toute perte de données en cas de problème.

• **Action** : Planifiez des sauvegardes automatiques hebdomadaires ou mensuelles pour sécuriser les données du site.



6. Lancement et suivi post-lancement

Formation des équipes sur l'utilisation du CMS

- **But** : Former les équipes à l'utilisation du CMS pour la gestion quotidienne des contenus et la maintenance.

- **Outils** :
  - **Team**, **Zoom** : Pour enregistrer et partager des tutoriels vidéo de formation.
  - **Confluence**, **Notion** : Pour créer et partager des guides de formation et des documents d'instruction.





7. Suivi continu et itérations

Suivi des performances et des retours utilisateurs

- **But** : Analyser les performances et recueillir les retours des utilisateurs pour identifier les pistes d'amélioration.

- **Outils** :
  - **Google Analytics**, **Hotjar**, **Crazy Egg** : Pour analyser le comportement des utilisateurs et générer des cartes de chaleur.
  - **Typeform**, **SurveyMonkey** : Pour collecter des retours utilisateurs par des enquêtes.
  - **FullStory**, **Lucky Orange** : Pour enregistrer les sessions des utilisateurs et comprendre les interactions en détail.



7. Suivi continu et itérations

Maintenance et mises à jour du CMS

- **But** : Assurer la maintenance régulière du CMS, appliquer les mises à jour de sécurité, et ajouter des fonctionnalités au besoin.

- **Outils** :
  - **ManageWP**, **InfiniteWP** : Pour la gestion des mises à jour et la maintenance des sites WordPress.
  - **Composer** (Drupal), **WP-CLI** (WordPress) : Pour automatiser les mises à jour du CMS et des plugins.
  - **Sucuri**, **WordFence** : Pour assurer la sécurité et surveiller les vulnérabilités.



4. Production & Déploiement

8. Révisions et améliorations

Mesurez et itérez

Buts

- Surveillez les données d'utilisation et continuez d'améliorer le produit.
- Les concepteurs peuvent revoir leur interface utilisateur en fonction des commentaires des utilisateurs pour améliorer l'expérience utilisateur.
- Cette étape peut impliquer des ajustements de la mise en page, des modifications de la typographie, des changements de couleurs et des améliorations de l'accessibilité.
- Les concepteurs peuvent itérer ce processus jusqu'à ce que l'interface utilisateur soit optimale.

